The Evolution

When the Google box launched, we customized the XSLT to put our logo, header, footer and simple CSS on it, but left most of the default settings.

We meant to get around to revising it.

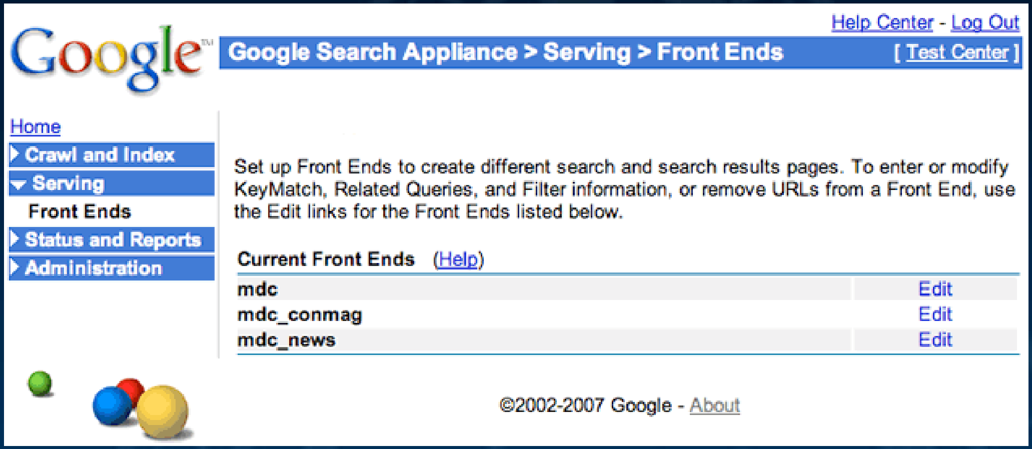
The Problem

We had old content from news releases and the magazine cluttering up our search results. Someone searching for “zebra mussels” would wade through a page of obsolete information before finding current information

- News from 1995-98
- Magazine articles from 1995
- We link extensively to our magazine archives as support material.
- Our staff uses the web to search for old articles by author, title or subject matter.

The Solution

Create separate collections for news and magazine articles.



Set up Front Ends to create different search and search results pages. To enter or modify KeyMatch, Related Queries, and Filter information, or remove URLs from a Front End, use the Edit links for the Front Ends listed below.

Current Front Ends (Help)

mdc	Edit
mdc_conmag	Edit
mdc_news	Edit

©2002-2007 Google - About

This pulls out the old magazine and news articles from the search results.

This helps the search engine pull more appropriate pages from our core library of documents.

The Implementation

Now, users that used to find magazine articles in the search results can’t find them.

We needed to be able to bounce between the collections on the search page; preferably with a switch to let users select which collection they wanted.

Steps to a Solution

1. Clean the HTML output from the XSLT
   - This took the most time. Had to understand how Google’s XSLT works and where elements were referenced. Ctrl F is your friend.
2. Develop collection switch
   - Insert new XSLT elements/logic to support switching of collections
   - Use javascript to change values on the fly (basic DOM programming)

Cleaning Up the Output

View the source code of existing HTML output from XSLT

- Identify elements to get rid of (table layout, <font>)
- Identify elements to change (<b>, <i>, headings, separators)

Locate those elements in the XSLT file and change accordingly

- Make sure logic isn't changed or that you know what has changed
- Make the output as close to XHTML Strict as possible

Cleaning Up the Output - cont

Add any new features

- Import new CSS
- Header/footer variables (used during development)
- Follow a template (header, footer, content, main-content)
- Fieldsets and labels to forms
- Javascript functions

Currently can't make XHTML Strict compliant due to namespace

(There is a beta Google XSLT that supposedly will generate; however, still couldn't get it to validate)

Switching Collections

1. Create a variable to determine when to search other collections
2. The site parameter is used to select a collection
   - Add XSLT logic to display the field when necessary
   - Add XSLT logic to show the value when page loads
3. Add a new javascript function to change values on the fly
   - Cycle through all the appropriate forms on the page (top/bottom)
   - Change the proxystylesheet and client parameters to the selected value

Breakdown of the Code - XSLT

Changes made in the XSLT

Created a new variable search_other_parts, that will trigger when to display the options of changing collections.

```
<xsl:variable name="search_other_parts">1</xsl:variable>
```

Added logic to form_params template to generate the site field if the search_other_parts is not activated.

```
<xsl:if test="$search_other_parts!='1'">
    <xsl:if test="$search_collections_xslt = '' and PARAM[@name='site']">
      <input type="hidden" name="site” value="{PARAM[@name='site']/@value}"/>
    </xsl:if>
  </xsl:if>
```

Breakdown of the Code - XSLT

Code was added to the search_box template to manually create the site field in the search form if needed.

First, check if the search_other_parts variable is turned on and that we are not searching through results.

Next, check the current value of site so it can be displayed when the page loads.

MDC online
MISSOURI DEPARTMENT OF CONSERVATION

CUSTOMIZING GOOGLE

```
<xsl:if test="($search_other_parts != '0') and ($type != 'swr')">
 <p>
  <xsl:choose>
    <xsl:when test="PARAM[(@name='site') and (@value='mdc_conmag')]">
      <label><input type="radio" name="site" value="mdc_conmag"
                onClick="javascript:changeStyle('mdc_conmag')" checked="checked" />Search
                Conservationist</label>
    </xsl:when>
    <xsl:otherwise>
      <label><input type="radio" name="site" value="mdc_conmag"
                onClick="javascript:changeStyle('mdc_conmag')" />Search Conservationist</label>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:choose>
    <xsl:when test="PARAM[(@name='site') and (@value='mdc')]">
      <label><input type="radio" name="site" value="mdc" onClick="javascript:changeStyle('mdc')"
                checked="checked" />Search MDC online</label>
    </xsl:when>
    <xsl:otherwise>
      <label><input type="radio" name="site" value="mdc" onClick="javascript:changeStyle('mdc')"
                />Search MDC online</label>
    </xsl:otherwise>
  </xsl:choose>
 </p>
 </xsl:if>
```

Breakdown of the Code - Javascript

Code added to the search_results template

The function is called and passed a variable when the radio button is selected.

The function cycles through all the forms found on the page and will only change forms containing both proxystylesheet and client

```
function changeStyle(theValue){
    for (var i=0; i < document.forms.length; i++) {
    if (document.forms[i].elements['proxystylesheet'] && document.forms[i].elements['client'])
    {
        document.forms[i].elements['proxystylesheet'].value = theValue;
        document.forms[i].elements['client'].value=theValue;
     }
}
```

Final Thoughts

There were 3 XSLTs, one for each collection. The changes outlined here were made to all three of these.

This example worked with relative ease due to the naming convention. Our client, proxystylesheet, site all have the same names/values: mdc, mdc_news, mdc_conmag. Same names allowed for easy switching of values.

If Javascript is disabled the collections will still be searched; however, the page design will not change.